NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

BSc THESIS

Pathlet Learning for Compressing and Planning Trajectories

Christos T. Tsapelas

Supervisor: **Dimitris Gunopulos,** Associate Professor

ATHENS

OCTOBER 2020

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εκμάθηση Συνόλου Μονοπατιών για τη Συμπίεση και Σχεδίαση Τροχιών του οδικού δικτύου

Χρήστος Θ. Τσαπέλας

Επιβλέπων: **Δημήτριος Γουνόπουλος,** Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2020

BSc THESIS

Pathlet Learning for Compressing and Planning Trajectorries

Christos T. Tsapelas
S.N.: 1115201500161

SUPERVISOR: **Dimitris Gunopulos,** Associate Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εκμάθηση Συνόλου Μονοπατιών για τη Συμπίεση και τη Σχεδίαση Τροχιών του οδικού δικτύου

Χρήστος Θ. Τσαπέλας

Α.Μ.: 1115201500161

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Δημήτριος Γουνόπουλος,** Καθηγητής

ABSTRACT

The last years, there has been a wide spread of GPS-enabled devices, which has led to the generation of unprecedented amounts of spatio-temporal trajectory data. These datasets offer a great opportunity for enhancing our understanding of human mobility patterns, thus benefiting many applications from location-based services.

In this work, we faced the problem of pathlet learning to understand shared structure in a large collection of trajectory data. The aim of this project is to extract a pathlet dictionary able to reconstruct all the input trajectories of the dataset, which can be later used as a base for route planning and travel time estimation.

Our methodology describes the process of mapping the GPS data to road segments on the map and proposes the definition of the problem as a problem of Linear Programming. Then, due to the complexity of the problem and the lack of the huge amount resources required, we implemented some preprocessing in the input trajectories which resulted in solving smaller problems. Finally, the results of our work are presented.

SUBJECT AREA: Learning on Spatio-Temporal Data

KEYWORDS: GPS trajectories, linear programming, pathlet learning, pathlet planning

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια, έχει υπάρξει μεγάλη διασπορά συσκευών εξοπλισμένων με λειτουργία εντοπισμού θέσης, γεγονός το οποίο έχει οδηγήσει σε μία πρωτοφανή παραγωγή γεωχωρικών δεδομένων που αναπαριστούν τροχιές στο οδικό δίκτυο. Αυτά τα σύνολα δεδομένων προσφέρουν μία σπουδαία ευκαιρία για να βελτιώσουμε την αντίληψη μας σχετικά με τα πρότυπα ανθρώπινης κινητικότητας, έτσι ώστε να επωφεληθούν πολλές εφαρμογές από υπηρεσίες βασισμένες στην τοποθεσία.

Σε αυτή την εργασία, αντιμετωπίσαμε το πρόβλημα εκμάθησης μονοπατιών με σκοπό να κατανοήσουμε κοινά μοτίβα σε μία μεγάλη συλλογή από δεδομένα τροχιών οδικού δικτύου. Σκοπός της εργασίας είναι να εξάγουμε ένα ευρετήριο μονοπατιών ικανό να ανακατασκευάσει όλες τις τροχιές εισόδου του συνόλου δεδομένων, το οποίο μπορεί αργότερα α χρησιμοποιηθεί σαν βάση για τον σχεδιασμό διαδρομών και αργότερα στην εκτίμηση χρόνου άφιξης.

Η μεθοδολογία που ακολουθήθηκε, περιγράφει την αντιστοίχιση των δεδομένων που προέρχονται από συσκευές εντοπισμού θέσης σε δρόμους του οδικού δικτύου και την διατύπωση του προβλήματος, ως ένα πρόβλημα Γραμμικού Προγραμματισμού. Στη συνέχεια, λόγω της πολυπλοκότητας του προβλήματος και της έλλειψης πόρων, εφαρμόστηκε μία επεξεργασία στα δεδομένα των διαδρομών, η οποία οδήγησε στην επίλυση πολλών μικρότερων προβλημάτων. Στο τέλος, παρουσιάζονται τα αποτελέσματα της εργασίας σχετικά με το μέγεθος των ευρετηρίων, το μέγεθος των εξαγόμενων μονοπατιών και ιστογράμματα μήκους των μονοπατιών.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εκμάθηση σε Χωροχρονικά Δεδομένα

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Διαδρομές GPS, γραμμικός προγραμματισμός, εκμάθηση μονοπατιών, σχεδίαση μονοπατιών

To my family

AKNOWLEDGMENTS

I would like to thank my supervisor, Prof. Dimitrios Gunopulos, and his PhD student, Nikolaos Zygouras, for giving the opportunity to work on this subject and helping me with their guidance and useful advice on obstacles that came up while implementing this thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

The work for this thesis was conducted between July 2019 and October 2020 at the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens and was supervised by the Prof. Dimitrios Gunopulos. The project was developed on a Linux machine using the Python programming language. For the development of the project it was a great importance to understand basic concepts and of Linear Programming and get familiar with the pulp library, a complete suite of solvers for this kind of problems

1. INTRODUCTION

The wide spread of GPS enabled devices has helped to create enormous datasets of trajectories of all available means of transport, such as, vehicles, pedestrians public transport etc. These datasets hold rich information about mobility data and provide the opportunities to improve our understanding about pedestrian's movement patterns in the network, vehicular traffic patterns, as well as they capture the state of the city road networks. Algorithms that analyze these data can be very helpful in many applications, such as route planning and transportation systems.

Some recent studies encouraged us and especially the work of Gonzalez et al. [6], who discovered that human trajectories show a high degree of spatial and temporal regularity and that humans, show high probability of repeating similar mobility patterns. In this work we intend to leverage some shared structure among human trajectories. For our work we have chosen the use of pathlets, which are frequently traversed path segments of the road network, to represent this spatial and temporal regularity.

1.1. Importance and challenges of problem

The problem of pathlet learning was formulated as solving an optimization problem, whose objective function minimizes the size of the pathlet dictionary as well as the number of pathlets used to reconstruct each trajectory. The extracted pathlet dictionary can benefit a variety of applications in various of ways. For example, this dictionary can be considered as a compressed form of the dataset in a manner that, by storing only the indices of the pathlets, we are able to reconstruct all input trajectories. Another application is the route planning field, because the pathlets encapsulate patterns of how people tend to travel in cities' road networks. This approach can give better results from existing algorithms, in problems like shortest path search.

However, we intend to find common mobility patterns, such as most frequently used road segments, our application focuses only in the first level of approximation of

common structures. Firstly, we focus only to find common patterns of use of mobile traffic and we do not capture information about the periodicity of data. For example, even we extract the information that central highways show high frequency of usage, we exclude the time of the day or the day itself the measurement took place. For example, every morning because the majority of the population travel to their job places, the traffic jam is significantly increased, which can make the drivers to think an alternate route in order to be on time to their work. Moreover, in weekends or in public holidays the mobility patterns behavior changes significantly.

Secondly, as every device used for measurements, the GPS devices also have a built-in error in their measurements. The initial data, as produced from the GPS device used, is a pair of longitude and latitude containing a lot of noise. This pairs must be matched to the road network to have some value. After the mapping is finished and the road segments are extracted, then an index is assigned to each road segment and each trajectory is represented as a set of indices.

1.2. Related Work

For the problem of trajectory segmentation, many approaches have been proposed. Zygouras and Gunopulos[2] have proposed the notion of corridors, which are the most frequent paths in a trajectory database. As corridor is defined a trajectory which connects two coordinates through a path that is frequently used. To discover these corridors, the input trajectories are grouped into clusters, where each cluster contains trajectories with similar movement. Then, for each road segment or segments, a score is calculated which depicts the likelihood of movement in this road segment or group of road segments. The higher the score, the most likely this path is frequently visited. The most frequent paths then are extracted, forming the desire set of corridors.

Gudmundsson et al.[7] propose a “bag of segments” method by splitting trajectories into small pieces and then aggregating them into clusters. This approach lacks in the variety of results, in a way that this method can detect segments of fixed length.

This thesis is based on paper **Pathlet Learning for Compressing and Planning Trajectories,** by Chen et al., in which the aim is to learn the most frequent subpaths in the dataset of different size, able to reconstruct all the trajectories given. Two algorithms are proposed in this work, the *Joint Approach* and the *Distributed Approach,* to extract the pathlet dictionary. In my project I implemented the first algorithm with some modifications, due to lack of resources. The dataset used in Chen's work, is produced from taxi trajectories in the city of Beijin. Instead, in my work a dataset of trajectories in the city of Porto is provided.

1.3. Objective

The main objective of this thesis is to extract a pathlet dictionary for the city of Porto by solving an optimization problem via linear programming. The project is implemented in python 3.8 accompanied with various python libraries. For the solution of the objective function the *pulp* library was used with the provided solvers. As for the data visualization stage, *matplotlib* (plotting library for Python, for the trajectories visualization *gmplot* (plotting library for Python using Google Maps) and *seaborn* (Python data visualization library based on matplotlib) were used to construct different plots and diagrams.

2. DATASET

This section summarizes the content of the dataset that was used and presents interesting data visualizations to provide a first look on the data.

2.1. Porto Dataset

2.1.1. Description

The provided dataset of Porto consists of two files. The *barefoorPolylinesPorto.cpickle* which contains all road segments used in the input trajectories. The pickle format allows us to manipulate this file as a dictionary where, the keys of the dictionary are the indices of the road segments and for each segment, the value is a list containing the pairs of longitude latitude as measured from the GPS device. The second file, called *mergedPorto.csv,* contains all input trajectories, where each trajectory is represented by an id and a list of indices pointing to the previously mentioned file. The dataset is consisted of approximately 150k trajectories, produced by 42.4k different row segments and it consisted of 1.85m rows.

2.1.2. Data Visualization

- **Trajectories**
- **An instance of a trajectory in Porto's downtown**.

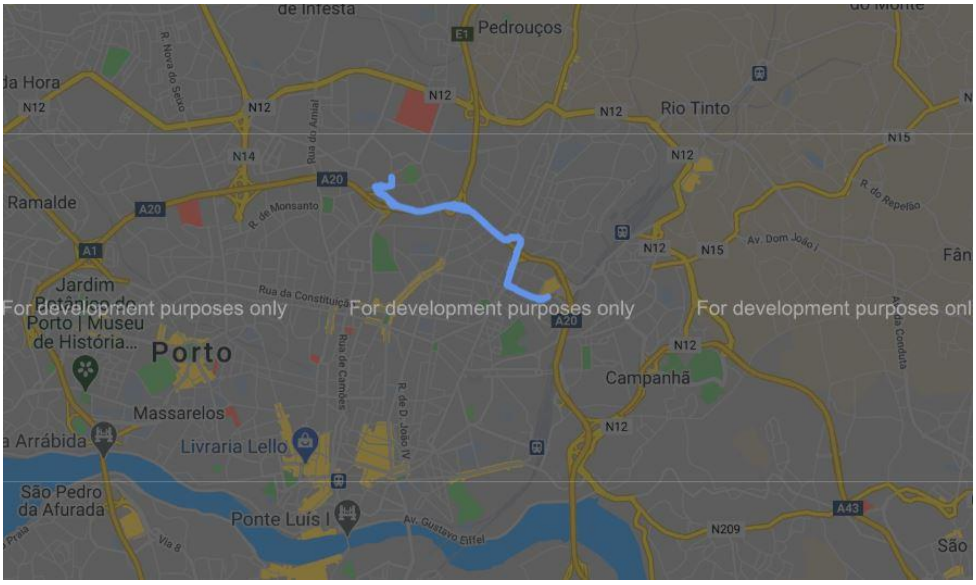

Figure 1: Line 1240 in Porto's downtown

- **Plot of all used road segments in Porto area.**

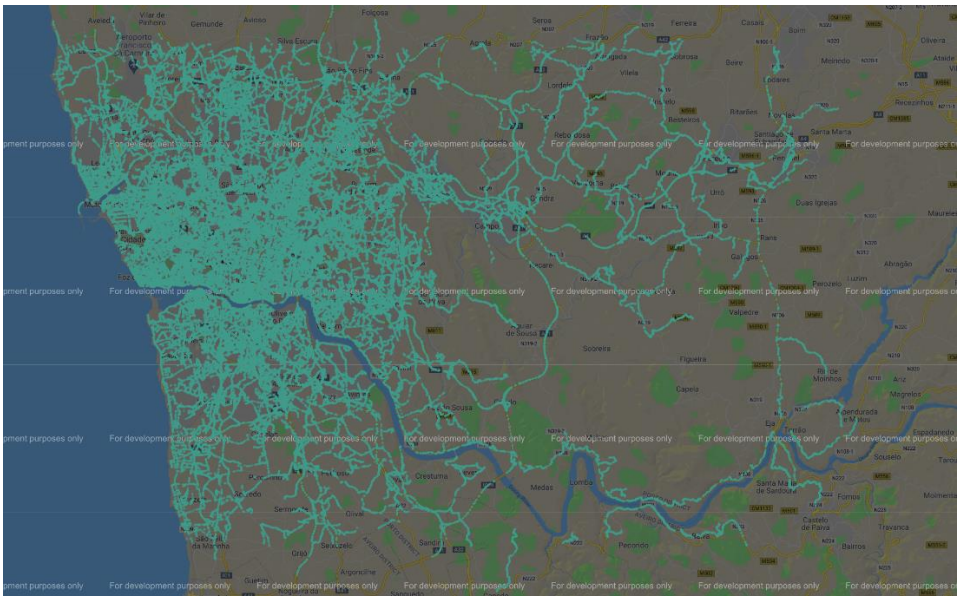

Figure 2: Usage of road network

From the above plot we can observe there is a big coverage of the road network of Porto reaching also its suburbs and some segments outside of this region.

2.1.3. Data Information

In this part we show some statistical information about the dataset.

- **Distribution of trajectory length**

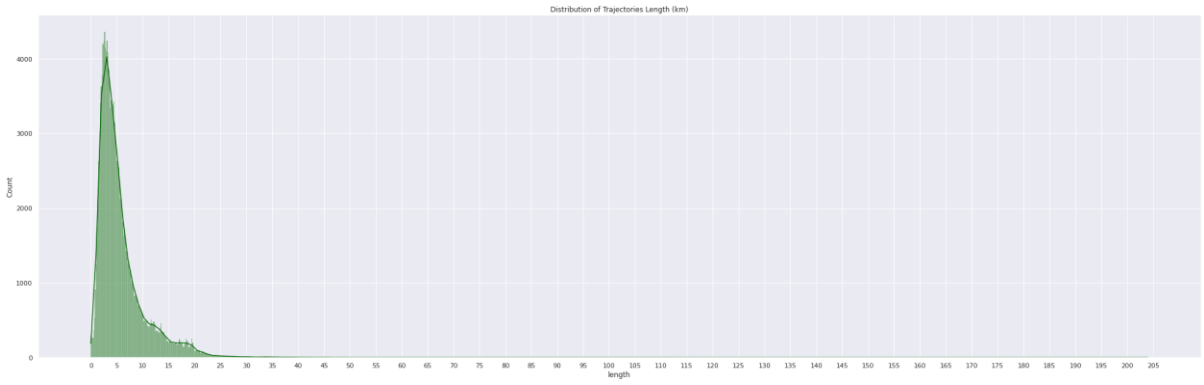

Figure 3: Trajectory length distribution

We observe the distribution of the length of input trajectories is close to the normal distribution, but there are a few too big outliers. The mean length is μ = 5.79km and standard deviation σ = *4.9*

- **Distribution of number of road segments in a trajectory**

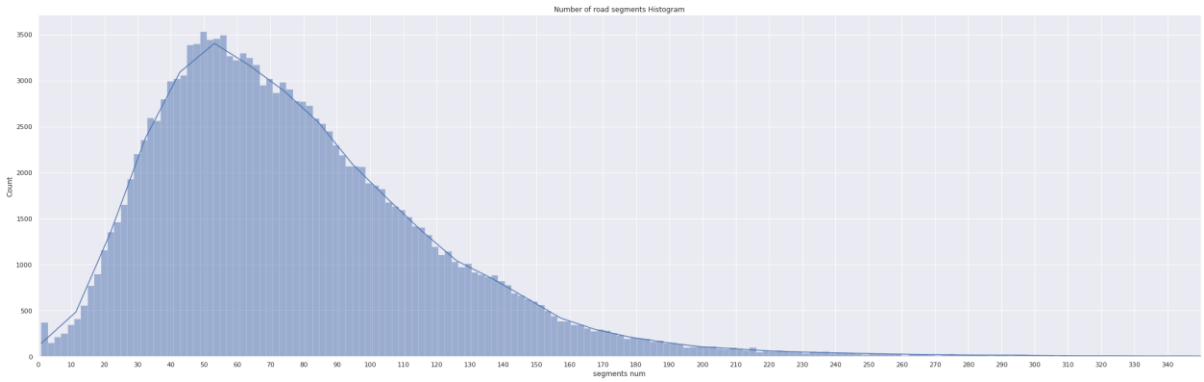

Figure 4: Road segment usage distribution

We observe here also, the usage of road segments is very close to the normal distribution. The input trajectories averagely use μ = 78,1 road segments with standard deviation σ = *45.07.*

- **Distribution of road segment' mean length**

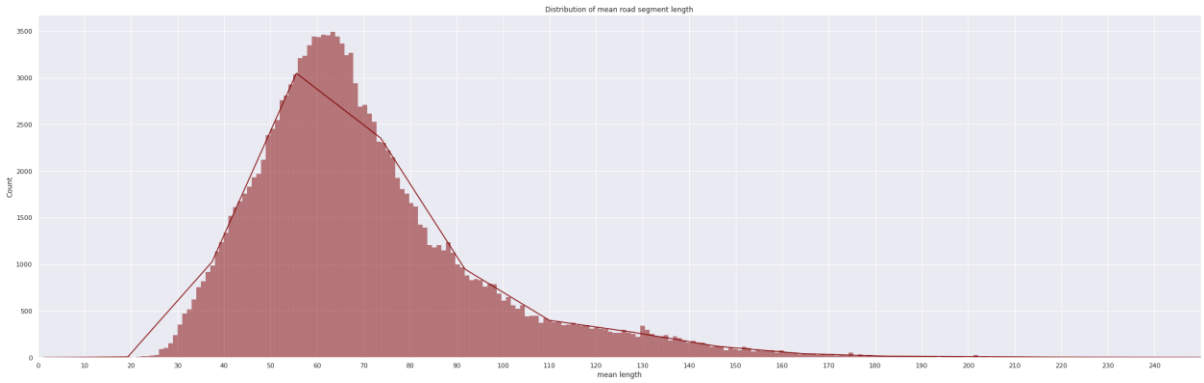

Figure 5: Road segment' mean length distribution

In the above plot, we show the distribution of road segments' mean length. This plot shows the behavior of the road segments' mean length each trajectory use. Again, this

plot is very close to the normal distribution with mean value $\mu = 70.69m$ and standard deviation $\sigma = 29.41m$.

- **Histogram of top 150 most used pathlets**

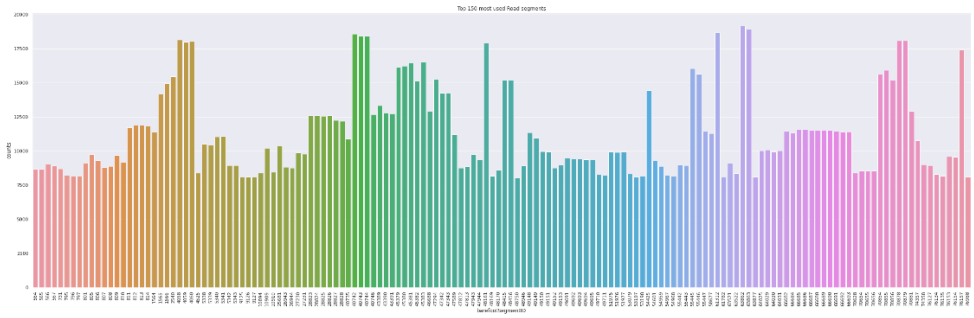

Figure 6: Most 150 used road segments histogram

And in the next plot we show where these roads are located. We observe these road segments are located downtown of Porto which makes sense. The most traffic is observed generally in cities' centers.

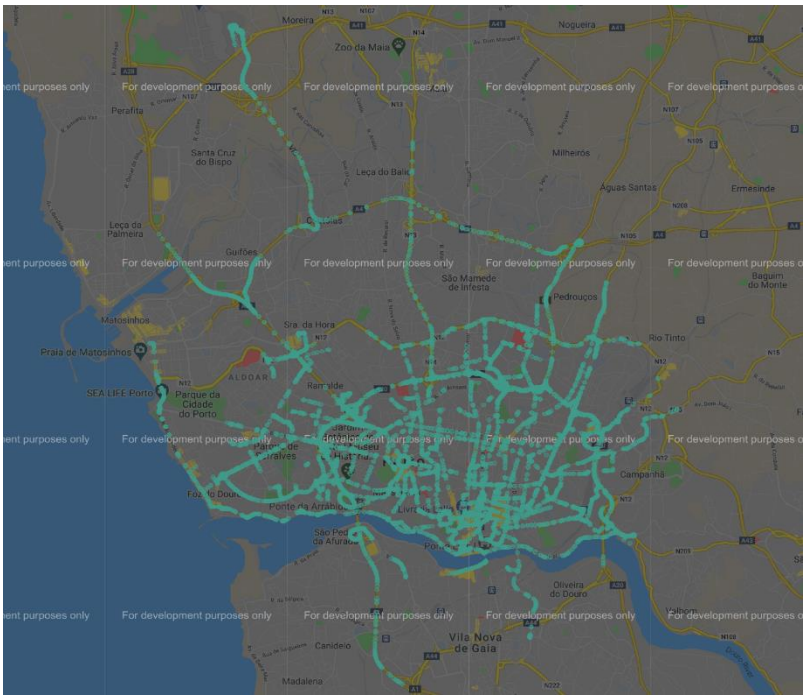

Figure 7: Top 3000 most used road segments

3. PROBLEM STATEMENT

In this section the optimization problem for pathlet learning from a collection of trajectories is formally defined. Firstly, some terms and definitions are introduced and then, the problem statement. The statements below can be applied to any city's road network.

- **Terminology**

We suppose the underlying network is formulated as a directed graph *G* = <*V, E*>, where *V* and *E* represent the set of nodes and edges, respectively. We define as *T* the collection of trajectories and each trajectory *t* in the collection, is a path on *G.* We call path *p* on *G* a *pathlet,* if it is a sub-path of one or more trajectories in *T.* As $P(t)$ we denote the set of all subpaths of a trajectory *t* and foreach one of these sets we can infer the following: $|P(t)| = \frac{|t|(|t|+1)}{2}$ where $|t|$ is the number of edges contained in *t.* Next, we denote the set of pathlets from all input trajectories as $\bar{P} = \bigcup_{t\epsilon T} P(t)$.

- **Definitions**

Definition 1. Each trajectory *t* can be reconstructed by a pathlet set $P = \{p_1, \dots, p_m\} \subset \bar{P}$, denoted by $t = u(P)$, if there is a permutation *σ* such that:

$$t = p_{\sigma(1)} p_{\sigma(2)} \dots p_{\sigma(m)}.$$

Easily we can understand that there may more than one possible permutations to reconstruct *t* using subsets of P. In this work we focus on permutations with minimum usage of pathlets..

Definition 2. We define the *description length dl(t, P)* of a trajectory *t,* with respect to a pathlet set P is given by:

$$dl(t,P) = \begin{cases} \min_{P_{sub} \subset P, t = u(P_{sub})} u(P_{sub}), & \exists P_{sub} \subset P, t = u(P_{sub}) \\ \infty & otherwise \end{cases}$$

The above definition denotes we seek the permutation with the minimum cardinality.

Definition 3. The *description length* of a *pathlet set* P is defined as:

$$dl(P) = \sum_{p \in P} |p|$$

- **Problem Statement**

As mentioned before, the goal is to minimize the cardinality of the description length of the extracted pathlet set as well as the description length for each trajectory. Given a collection of trajectories *T* and the set of all possible pathlets $\bar{P}$, we define pathlet leaerning as solving the optimization problem shown below:

$$\min_{P \in \bar{P}} dl(P) + \lambda \sum_{t \text{ in } T} dl(t, P)$$

$$s.t.\ \forall t \in T, \exists (P_{sub}) \in : t = u(P_{sub})$$

where λ parameter determines the tradeoff between these two terms. More specific, increasing λ value would reduce the number of pathlets that are used to reconstruct each trajectory, but enlarge the cardinality of extracted pathlet set. By default, the value of λ = 1.

4. ALGORITHM

In this section we describe the algorithm of Join Setting to solve the above optimization problem. This algorithm formulates the pathlet learning as an integer program and solves it by its linear programming relaxation.

4.1. Algorithm in the Joint Setting

The first objective is to convert the formulation mentioned above to an equivalent integer program with constraints.

- **Constraints and objectives**

For each pathlet $p \in \bar{P}$ we create a binary variable $x_p \in \{0,1\}$, where $x_p = 1$ if $p \in P$ and $x_p = 0$, otherwise. This definition basically assigns each pathlet a binary indicator and indicates if a certain pathlet is chosen to be part of the final pathlet dictionary. After defining these indicators, the description length of can be expressed as:

$$dl(P) = \sum_{p \in P} |p| = \sum_{p \in \bar{P}} |p| x_p$$

- **Reconstruction Constraint**

We desire to be able to reconstruct every trajectory $t \in T$ with a subset of the learned pathlet set *P,* thus we associate each pathlet $p \in P(t)$ with a binary variable $x_{t,p} \in \{0,1\}$ where $x_{t,p} = 1$ if pathlet p is used to reconstruct trajectory *t* and $x_{t,p} = 0$ otherwise. After the two binary variables are defined, if a pathlet *p* is chosen then the below inequality holds:

$$x_{t,p} \leq x_p, \forall p \in P(t), t \in T$$

- **Decomposition constraint**

In order to extract the minimum pathlet dictionary, the extracted pathlets in a trajectory should not be overlapped. In order to achieve this we introduce the below constraint:

$$\sum_{p \in P(t),\ e \in p} x_{t,p} = 1, \quad ,\forall e \in t, t \in T$$

In other words, the above constraint ensures that for all pathlets $p \in P(t)$ of a trajectory *t,* each edge of the trajectory belongs to one and only one pathlet. Thus, the overlaps of pathlets are avoided and the reconstruction can be efficiently done.

- **Integer Programing**

Inserting the above constraints in the problem's equation, the integer program is defined as follows:

$$\min \sum_{p \in \bar{P}} x_p + \lambda \sum_{t \in T} \sum_{p \in P(t)} x_{t,p}$$

$$s.t. \quad x_{t,p} \leq x_p, \forall p \in P(t),\ t \in T$$

$$\sum_{p \in P(t),\ e \in p} x_{t,p} = 1, \quad ,\forall e \in t, t \in T$$

$$x_p \in \{0,1\}, \quad \forall p \in \bar{P}$$

$$x_{t,p} \in \{0,1\}, \quad \forall p \in P(t), t \in T$$

5. MEMORY PROFILING

In this section the memory consumption of a trajectory is calculated, containing the mean number of road segments, as calculated above. Based on these measurements, we make an approximation of memory usage and how we solved the lack of resources problem by splitting the dataset into a grid of cells. The main concept is to solve many smaller problems.

5.1. **Memory profiler output**

We plot the profiler output for two key points of our implementation:

- Memory profiling on creation of *P(t)*

Figure 8: Creation of pathlet set P(t) for a trajectort t

- Decomposition constraint for a trajectory *t*

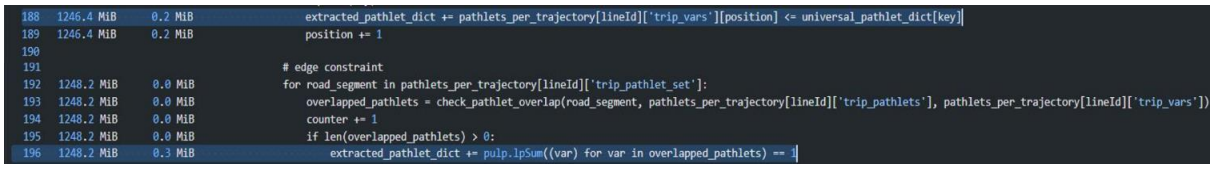

Figure 9: Creation of decomposition constraints

The highlighted lines in figure 8, output the memory consumption of the creation of pathlet variable $x_{t,p}$, which is the same amount of memory for the unique indicators x_p. From the definition of the cardinality of a pathlet set *P(t),* and more specifically $|P(t)| = \frac{|t|(|t|+1)}{2}$, for a trajectory containing the mean amount of road segments, the cardinality is equal to $|P(t)| = \frac{|t|(|t|+1)}{2} = \frac{70 \cdot 71}{2} = 2.485$ pathlets. Thus, the amount of memory required only for both categories of variables of integer problem is

approximately $2.485 \cdot 0.2 \cdot 2 \cong 996mb$. Next, the first highlighted line in figure 9, output the memory consumption about the constraint: $x_{t,p} \leq x_p$, and the second one outputs the decomposition constraint:

$$\sum_{p \in P(t),\ e \in p} x_{t,p} = 1, \quad ,\forall e \in t, t \in T$$

In terms of memory consumption, the first constraint translate to $2.485 * 0.2 = 449mb$ and the second one is related to the number of road segments, thus #road segmetnts * memory consumption = $70 \cdot 0.3$ = 21*mb*. Summing up, the memory needs of a trajectory *t* is: variables creation + variables constraints + decomposition constraint = 996*mb* + 449*mb* + 21mb = 1466*mb* = 1,466*gb.* For the whole collection of trajectories the total amount of memory approximately needed is: #trajectories * memory per trajectory = 150.000 * 1.466*gb* ≅ 219.9*tb.* It is obvious that the amount of memory needed is enormous and it is rational, because we are creating all possible sub-trajectories which results in million constraints.

5.2. Splitting Dataset – Grid application on map

In order to be able to solve the pathlet learning problem, we applied a grid over the Porto area. Then each trajectory's is splitted into pieces and each sub-trajectory is assigned to the cell which geographically belongs. Next, we show graphically this procedure. The grid has length approximately 50km and 34km width, covering an area of 1520km^2. Each cell of the grid has 2,2km length and width respectively. There are 400 total cells in the grid.

- **Grid over Porto**

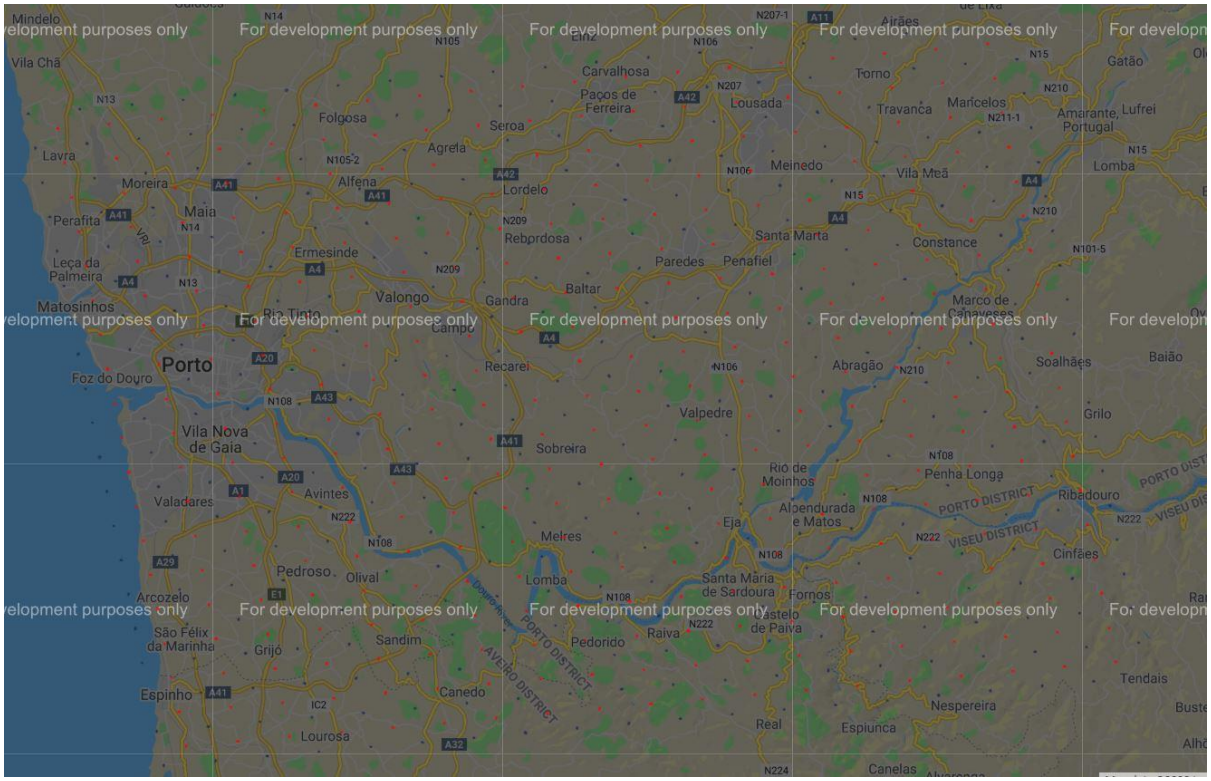

Figure 10: Grid over Porto

The black dotes are for the boundaries of the cells and the red dots are the centers of each cell.

- **Trajectory split into pieces**

In order to split each trajectory into pieces, we traversed all the road segments that are contained in and then, we calculated the haversine distance between the middle point of the road segment and all the centers of the grid. Finally, each road segment is assigned to the cell with the minimum haversine distance, in other words the closest cell. Below the plots of trajectory in Porto's downtown and a group of trajectories in a cell, in Porto's downtown

- **Splitted Trajectory**

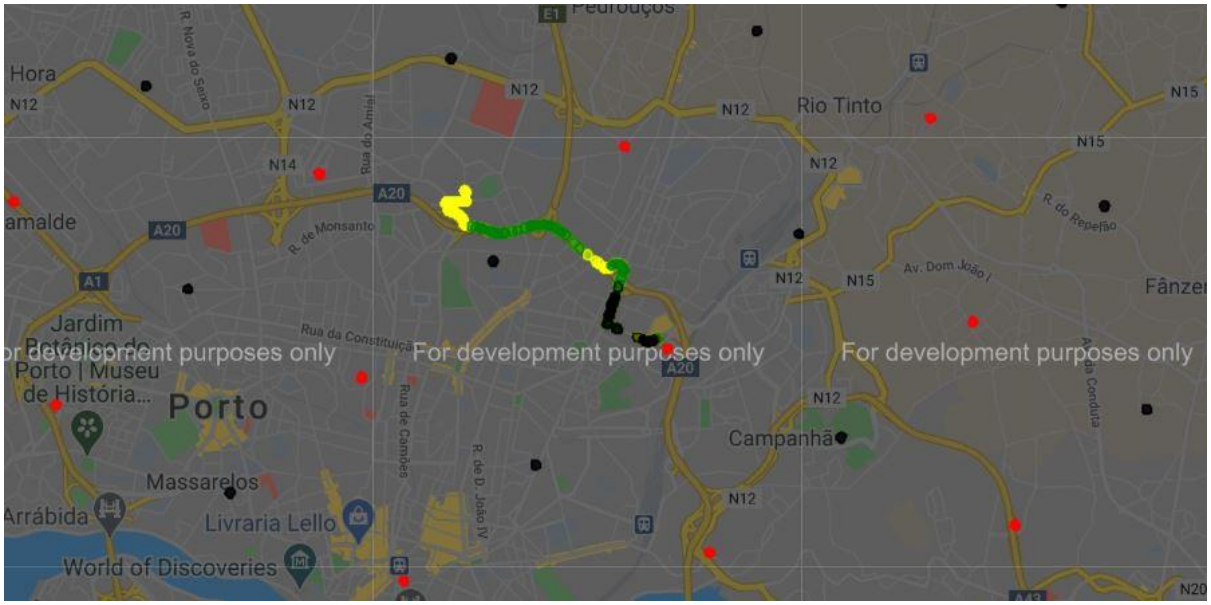

Figure 11: Splitted trajectory between cells

- **Trajectories of a cell**

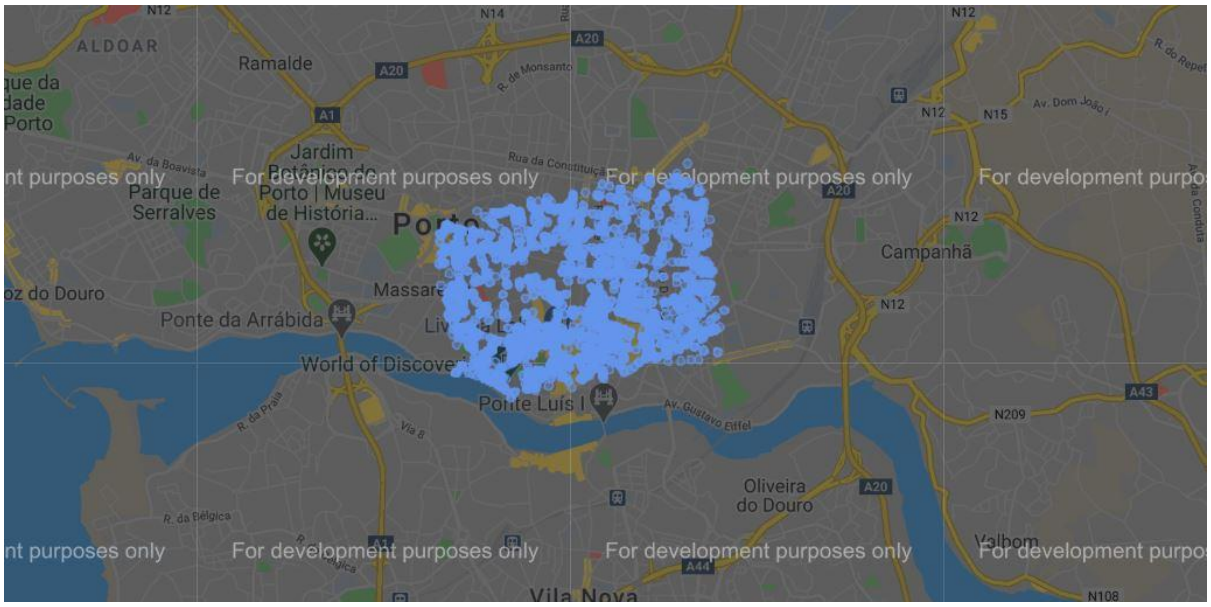

Figure 12: Sub-trajectories in a cell in Porto's downtown

6. RESULTS

In this section we show the results of our experiments. We show the importance of parameter lambda and how it affects the pahtlet dictionary size the number of road segments contained in pathlet. Then we show the distribution of pathlet length distribution in kilometers per λ.

- **Distribution of road segments usage per lambda**

- Dictionary size = 23.531

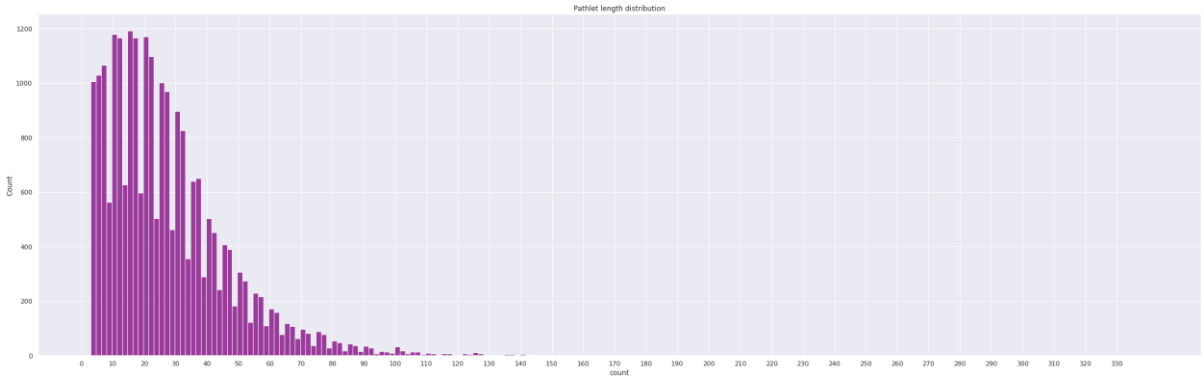

Figure 13: Pathlet length distribution λ = 1

- Dictionary size = 30.389

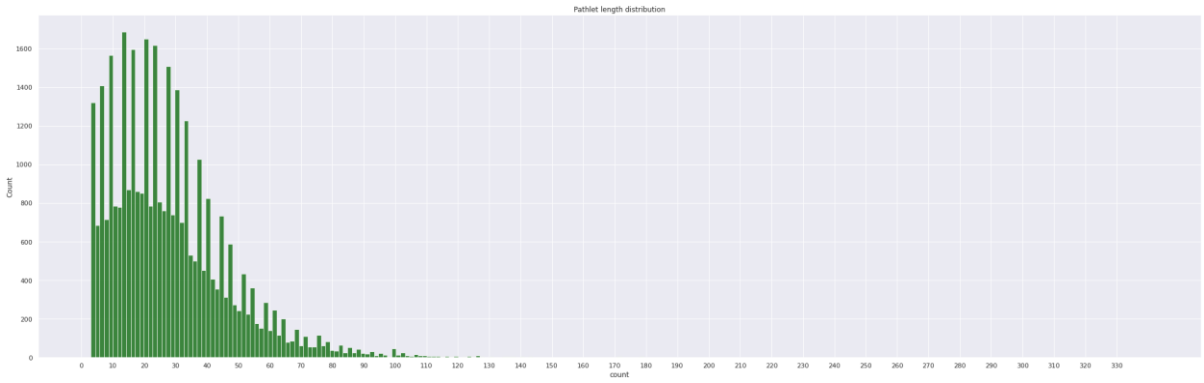

Figure 14: Pathlet length distribution λ = 10

- Dictionary size = 34.398

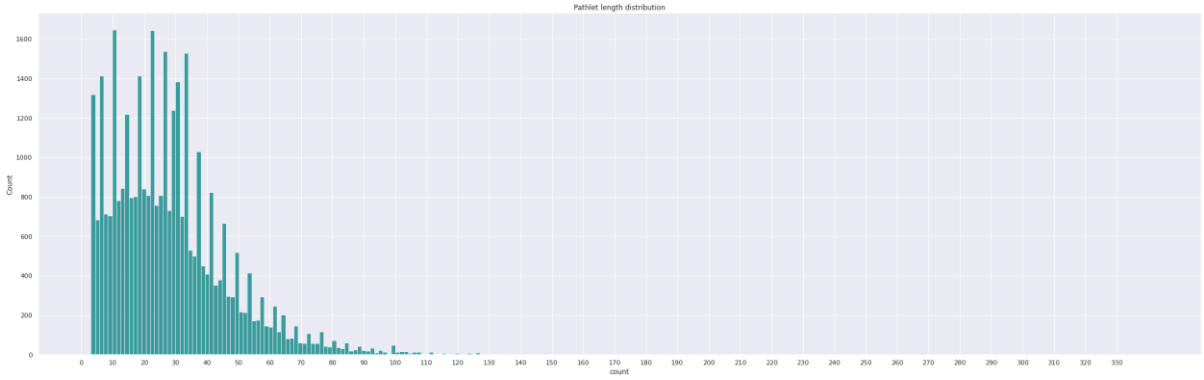

Figure 15: Pathlet length distribution λ = 100

- Dictionary size = 37.672

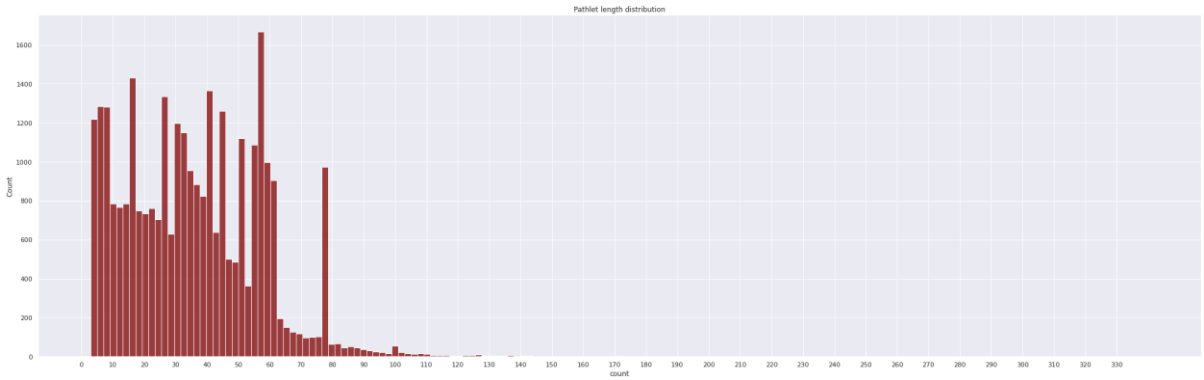

Figure 16: Pathlet length distribution λ = 1000

We notice that increasing the parameter lambda the dictionary size is increased as well as, the usage of rad segments is increased

- **Distribution of pathlet length in kilometers per lambda.**

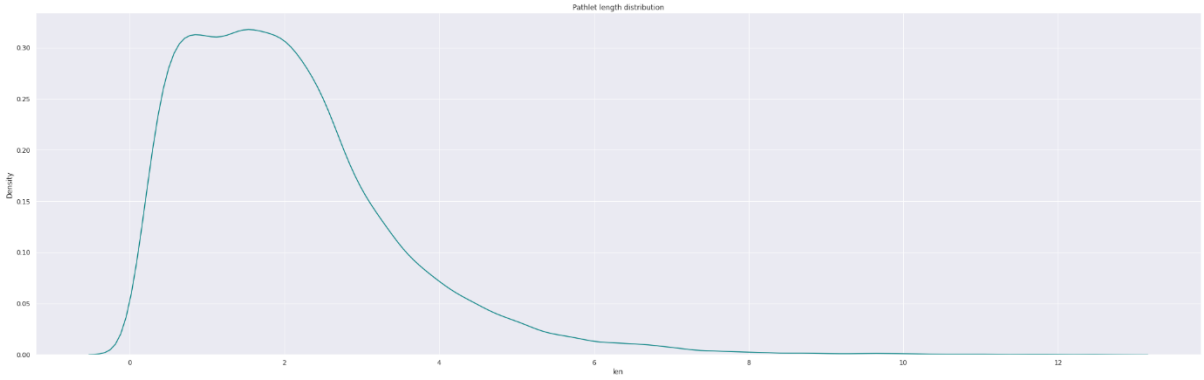

Figure 17: Pathlet length in kilometers λ = 1

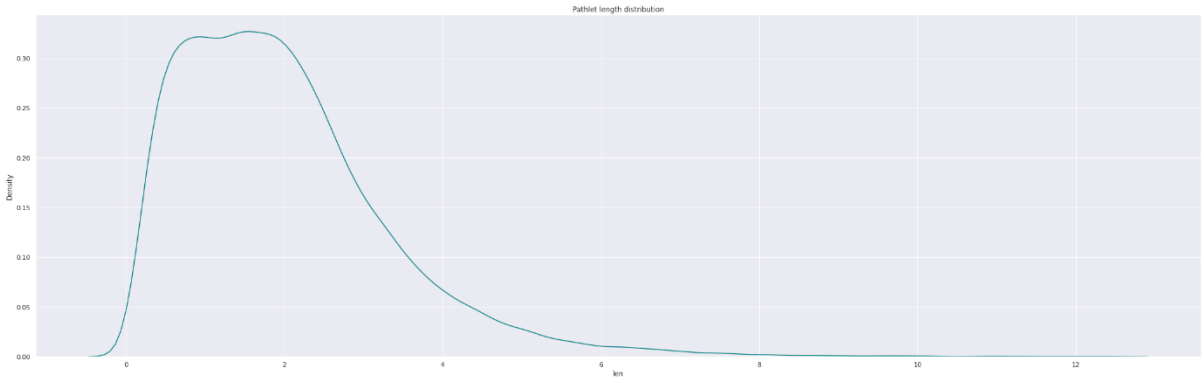

Figure 18: Pathlet length in kilometers λ = 10

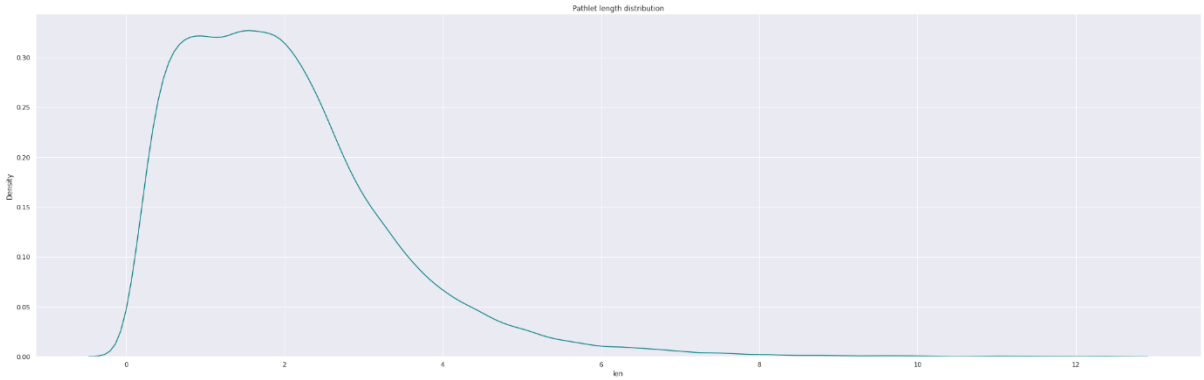

Figure 19: Pathlet length in kilometers λ = 100

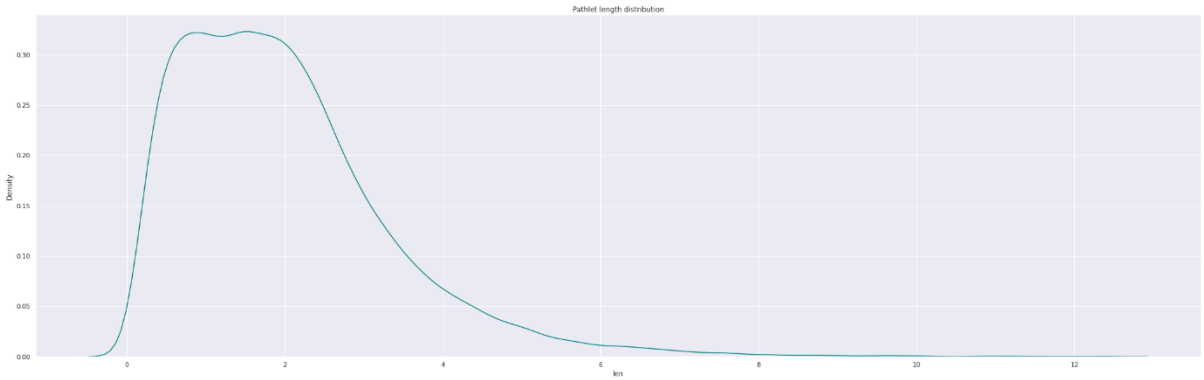

Figure 20: Pathlet length in kilometers λ = 1000

We notice the distribution on pathlet length does not changes significantly for the different values of lambda parameter.

- **Execution Times**

The cells were executed in pairs of 3 to save time. We show the average execution time per λ and the total time if they were executed sequentially. The last column is shown, because the number of trajectories and the number of road segments used, determines the difficulty of each problem. For some cells, the solver needs significantly more memory than others, thus they were executed separately.

Table 1: Execution Times per λ

λ	Mean time per cell(hours)	Total time (hours)
1	0.29	7.53
10	0.25	7.21
100	0.21	6.9
1000	0.2	6,4

7. CONCLUSION

7.1. **Discussion on Results**

The objective of this thesis was to reach the first level of approximation in learning frequent mobility patterns. We implemented the algorithm in the Joint Setting, which formulates the problem as problem of linear programming and we tested our implementation in large dataset of trajectories over Porto.

The results in the previous section has shown that the achieved compression of the dataset has reached a ratio of 33%. Moreover, the tradeoff between a small dictionary size and a dictionary with large pathlets, is shown above. In the first case, more pathlets are needed to reconstruct a trajectory, where in the latter, the trajectories this number decreases. On the other hand, we noticed the pathlet length distribution in kilometers has not changed significantly. We expected this result, because is an aftermath of the application of the grid over Porto and decomposition of the trajectories, in smaller trajectories in each cell it goes though.

7.2. **Future Work**

The pathlet learning approach can be helpful in applications such us travel time estimation. By defining pathlets of various number of road segments, has the advantages of capturing human mobility patterns and learning more complex traffic patterns. The link-based approaches, where the trajectories are reconstructed by a set of road segments only are used a lot, but lack on time estimation of the transition between road segments. A next step is to explore scalable methods to solve this linear program and investigate additional applications that can benefit from the result.

This project focuses on extraction of the pathlet dictionary by solving many similar smaller problems. An expansion to this approach is someone to experiment with different solvers, compare the results and explain the circumstances that a solver gives a better solution from another. Moreover, the comparison between the results of *Join Setting* and the *Distributed Setting,* as described in Chen, would give a better understanding on the pathlet approach. Finally, the second level of approximation is taking into consideration the periodicity of mobility patterns and how it changes the extracted pathlet dictionary

ABBREVIATIONS - ACRONYMS

CSV	Comma-seperated values
GPS	Global Positioning System

REFERENCES

[1]]Chen Chen, Hao Su, Qixing Huang, Lin Zhang, and Leonidas Guibas. 2013.Pathlet learning for compressing and planning trajectories. *In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* ACM, 392-395.

[2] Nikolaos Zygouras, Dimitrios Gunopulos: Discovering Corridors from GPS Trajectories. In *Proceedings of SIGSPATIAL'17, Los Angeles Area, CA, USA, November 7-10, 2017.*

[3] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers., January 2011.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[5] Maike Buchin, Anne Driemel, Marc van Kreveld, and Vera Sacrist´an. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 202–211. ACM, 2010*.

[6] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[7] Joachim Gudmundsson, Andreas Thom, and Jan Vahrenhold. Of motifs and goals: mining trajectory data. *In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, pages 129–138. ACM, 2012*.

[8] "Seaborn Documentation" [Online]. Available: https://seaborn.pydata.org/api.html

[9] "Gmplot Documentation" [Online]. Available: https://github.com/gmplot/gmplot/wiki

[10] "PuLP Documentation" [Online] Available: https://coin-or.github.io/pulp/

[11] Priyansh Soni "Linear Programming Using Python", towardsdatasciennce, 2019, [Online]. Available: https://towardsdatascience.com/linear-programming-using-python-priyansh-22b5ee888fe0